

trpr 2.0b1 User's Guide

Trpr (TRace Plot Real-time) is a program which analyzes output from the *tcpdump* packet sniffing program and creates output suitable for plotting. It also specifically supports a range of functionality for specific use of the *gnuplot* graphing program. *trpr* can operate in a "real-time" plotting mode where *tcpdump* `stdout` can be piped into *trpr* and *trpr*'s `stdout` in turn can be piped directly into *gnuplot* for a sort of real-time network oscilloscope. *Trpr* can also parse *tcpdump* text trace files and produce files which can be plotted by *gnuplot* or imported into other plotting or spreadsheet programs. IPv4 and IPv6 traces from *tcpdump* are supported. *Trpr* can also perform the same functions with *mgen* log files (See <http://mgen.pf.itd.nrl.navy.mil> for more information on *mgen* and the MGEN test tool set) and *ns-2* (Berkeley's network simulator - see <http://www.isi.edu/nsnam/ns>) trace files.

By default, *trpr* creates a "data rate" versus time plot of the flows specified using the *auto* and *flow* (and *exclude*) filtering commands. The *auto* command is used to set filters to automatically detect and *enumerate* individual flows matching the *auto* filter parameters (protocol type, source addr/port, and destination addr/port) and the *flow* command *aggregates* flows matching its filter specification under a single data plot set. The *exclude* command is used to specify packet flows *trpr* should ignore. The *flow*, *auto* and *exclude* commands can each be used multiple times on the command line to specify different combinations of filters to produce different desired output. (In the future, an exclusion filter set will also be provided).

If the *interarrival* command is used, *trpr* creates a plot of the differential interarrival delay of packets for the specified flows. And for MGEN packets, the *latency* command can be used to create a plot of the transmission latency (*mgen*-logged `rxTime - txTime`) versus time for the flows. Also, for MGEN packets, the *loss* command can be used to generate profiles of packet loss over time. MGEN packet payloads contain sequence numbers and time stamps to facilitate these analyses. The *count* command simply produces counts of the indicate "send" and/or "recv" events for the specified flows. The *histogram* command causes *trpr* to output histograms of any of these statistics and the *window* command determines the averaging window interval to use (with "`window -1`" over the entire trace file and "`window 0`" for individual events).

Trpr can also "play back" a *gnuplot* visualization of trace file content at real time rates with the *replay* command.

Other interesting (at least I find them interesting) options are planned for the future.

Downloads:

The *trpr* package is available at <http://downloads.pf.itd.nrl.navy.mil/proteantools>

Tcpdump can be found at <http://ee.lbl.gov>

Gnuplot's official web site is <http://www.gnuplot.org>

The *MGEN* web site is <http://mgen.pf.itd.nrl.navy.mil>

The *ns* web site is <http://www.isi.edu/nsnam/ns>

Build Instructions:

Simply compile *trpr* with a C++ compiler. It has been primarily built with *gcc* on Unix platforms. For example,

type:

```
g++ -o trpr trpr.cpp -lm
```

to build the executable binary.

Quick Start:

Here are a couple of examples illustrating use of *trpr* in simple ways. Note that *trpr* has a number of flexible command-line operations to get the results you want and understanding these is strongly recommended. And *tcpdump* has very flexible filtering options for paring down the data captured from the network so that your graphs can focus on the data of interest. The options of *tcpdump* and *trpr* can be coupled together in many different ways. And *trpr* supports options to command *gnuplot* to create Gif or Postscript files for hard output or use in other programs. Detailed usage instructions for *trpr* and hints for *tcpdump* and *gnuplot* usage are given later.

1) Non-real-time Operation:

- a) Capture IP packets with *tcpdump* with hexadecimal packet header output. Note you **must** use *tcpdump*'s hexadecimal output option (`-x`) and some form of filtering that captures only IP packets (*trpr* will not properly parse the output of non-IP data (e.g. Appletalk, etc) data which *tcpdump* may otherwise capture:

```
tcpdump -x ip <traceFile>
```

- b) Use *trpr* to process the captured `<traceFile>` to create a `<plotFile>` suitable for plotting with *gnuplot*, automatically creating lines on the graph for each unique "flow" of data discovered in the `<traceFile>`:

```
trpr input <traceFile> auto X output <plotFile>
```

- c) Use *gnuplot* to display a graph of *trpr*'s analysis results (By default *trpr* puts appropriate headers in the `<plotFile>` for *gnuplot*:

```
gnuplot -persist <plotFile>
```

As examples, *mgen* log files can be processed with:

```
trpr drec input <drecLogFile> auto X output <plotFile>
```

and *ns-2* simulation trace files can be processed with:

```
trpr ns input <nsTraceFile> link <srcNode>,<dstNode> send auto X output <plotFile>
```

Note: The link command coupled with the send command specifies to process packets sent over the link from node `<src>` to node `<dst>` in the *ns-2* simulation. The `<src>` and/or `<dst>` arguments can be wildcarded with the 'X' character to process multiple links to/from a particular or any simulation node.

Note: For *ns-2* mobile trace files, the link command should be used in the form:

```
link <nodeId>,{AGT | RT | MAC}
```

to capture the corresponding set of packets (Agent, Router, or MAC) for a mobile *ns-2* node).

We hope to provide more examples for using *trpr* with *ns-2* soon.

2) Real-time Operation:

- a) Set up *tcpdump* to capture IP packets and direct hexadecimal output to *trpr*, in turn piping *trpr*'s real-time output directly to *gnuplot* to get continuously updated plots of network traffic flow activity

Note you **must** use *tcpdump*'s hexadecimal output option (*-x*) and some form of filtering that captures

only IP packets (*trpr* will not properly parse the output of non-IP data (e.g. Appletalk, etc) data which

tcpdump may otherwise capture:

```
tcpdump -l -x ip | trpr real auto X | gnuplot -noraise -persist
```

Or for *mgen* operation:

```
mgen flush output /dev/stdout | trpr drec real auto X | gnuplot -noraise -persist
```

Note that the "tail -f" option can also be used to pipe a *mgen* log file to *trpr* in parallel with logging. (The *mgen* "flush" option causes *mgen* to "flush" its output line by line for better real time performance. Note this may penalize system performance)

Detailed Instructions:

Usage:

```
trpr    [version][drec][ns][raw][key][real][latency][interarrival][loss][count]
        [window <sec>] [history <sec>]
        [auto <type,srcAddr/port,dstAddr/port>,flowId]
        [flow <type,srcAddr/port,dstAddr/port>,flowId]
        [exclude <type,srcAddr/port,dstAddr/port>,flowId]
        [input <inputFile>] [output <outputFile>]
        [link <src>[,<dst>]][send | recv]
        [range [<startSec>][:<stopSec>]][offset <hh:mm:ss>]
        [summary][histogram][replay <factor>]
        [png <pngFile>][post <postFile>][multiplot][ramp][alimit <bytes>]
```

NOTE: Type, addr, or port parameters can be "wildcarded" with an 'X' character.

Command Line Parameters and Options:

version	Causes <i>trpr</i> to display program version number and exit.
drec	<i>trpr</i> will expect to process a <i>drec</i> log file instead of <i>tcpdump</i> hex output.
ns	<i>trpr</i> will expect to process a <i>ns</i> trace file instead of <i>tcpdump</i> hex output.
raw	When this option is given, the <outputFile> will only include unlabeled sets of plotting data without the default <i>gnuplot</i> compatible headers. This is useful to get the "raw" plot data

key	<p>for importing into a spreadsheet or other plotting program.</p> <p>With this option, <i>trpr</i> will print a "key" to the data plot sets in the <outputFile>. This consists of one comma-delimited line with a leading "#" character. This line is printed when new flows of data are detected and another data set column is output. The first column is marked "Time". Subsequent columns are labeled with a description of the flow data being plotted.</p>
rate	<p>Causes <i>trpr</i> to create plots of data rate versus time. The <u>window</u> command can be used to set <i>trpr</i>'s rate averaging window. The <u>rate</u> command is the implicit default plot mode for <i>trpr</i>.</p>
interarrival	<p>Causes <i>trpr</i> to create plots of differential interarrival packet delays for detected flows instead of the default data rate versus time plot.</p>
latency	<p>Causes <i>trpr</i> to create plots of transmission delay for <i>drec</i> flows instead of the default data rate versus time plot. This type of plot is only available for <i>drec</i> operation.</p>
loss	<p>Causes <i>trpr</i> to create plots of packet loss based on received sequence numbers for <i>drec</i> flows instead of the default data rate versus time plot. This type of plot is only available for <i>drec</i> operation. The <u>window</u> command can be used to set <i>trpr</i>'s loss averaging window. The "window" specified should be large enough to encompass several expected packet events for desired results.</p>
count	<p>Causes <i>trpr</i> to create plots of packet counts versus time instead of the default data rate versus time plot. The <u>window</u> command can be used to set <i>trpr</i>'s count accumulation window. The <u>rate</u> command is the implicit default plot mode for <i>trpr</i>.</p>
real	<p>When this option is given, <i>trpr</i> will output plotting commands and data to its <code>stdout</code>. This output is intended for the <code>stdin</code> of <i>gnuplot</i> for real-time plotting. However, note that this output can be redirected to a file for storage, and then later that file can be directed to the input of <i>gnuplot</i> for "playback". Note that the "real-time" mode can be used simultaneously with <i>trpr</i>'s cumulative "non-real-time" <u>output</u> option. Note that the "real time" graph update occurs once per <u>window</u> time. This option can also be used with pre-existing trace files. Use the replay command to limit the actual graph animation rate or the trace file will be parsed at "cartoon rate" (i.e. as fast as possible).</p>
gif <gifFile>	<p>This option commands <i>gnuplot</i> to create a "gif" (Graphics Interchange Format) file when it plots instead of the default X11 display. The <gifFile> parameter is the name of the file <i>gnuplot</i> will create when it processes <i>trpr</i>'s output. This can be used in either real-time or non-real-time operation. In real-time operation, the <gifFile> will be periodically overwritten according to <u>window</u> setting.</p>
post <postFile>	<p>This option commands <i>gnuplot</i> to create a Postscript file when it plots instead of the default X11 display. The <postFile> parameter is the name of the file <i>gnuplot</i> will create when it processes <i>trpr</i>'s output. This can be used in either real-time or non-real-time operation. In real-time operation, the <postFile> will be periodically overwritten according to <u>window</u> setting.</p>

multiplot	With <i>gnuplot</i> , <i>trpr</i> will create a "multiplot" graph with one graph per detected flow (stacked vertically). (This only works with the real-time updated (<u>real</u> command) graphing mode for now).
ramp	By default, <i>trpr</i> creates "stair step" plots of its averaging <u>window</u> results (i.e. 2 data points per window). The optional <u>ramp</u> command causes <i>trpr</i> to create plots with one data point per averaging window (at the window's end), thus "ramping" from one window to the next. This may be useful for alternative post-processing of <i>trpr</i> 's output files or to reduce the number of data points on plots with an extremely large number of data points where the <u>window</u> start/stop points are indiscernable anyway.
window <sec>	This parameter sets the step size of <i>trpr</i> 's window-based data rate and packet loss averaging algorithms. The step size unit is time in seconds. This algorithm counts the cumulative quantity of data (or packet loss) in each window of time and calculates the kilobits-per-second (kbps) (or loss fraction) value for each step. These discrete values of data rate (or loss fraction) versus time comprise <i>trpr</i> 's plot data. Two points are plotted, one at each time window's beginning and one at its end, to form a "stair step" plot. The <u>window</u> command also controls the <i>gnuplot</i> real-time graph update rate for <u>real</u> command operation. The <u>window</u> <sec> value can be specified as "-1" to cause <i>trpr</i> to average across the entire trace file (or the period specified by the <u>range</u> command). Note the negative <u>window</u> value should not be used in combination with the <u>real</u> command. Default = 1 second.
history <sec>	This parameter determines the range (in time units of seconds) of the X-axis of the graphs produced in <i>trpr</i> 's real-time mode. As time progresses, the <i>gnuplot</i> graphs will scroll in "strip-chart" fashion to display the current <u>history</u> of network activity. Default = 20 seconds.
auto <type,srcAddr:port,dstAddr:port,id>	This command instructs <i>trpr</i> to automatically discover and plot "flows" of network data according to the matching (type,src,dst,id) criteria provided. Otherwise, <i>trpr</i> only plots "flows" given by the <u>flow</u> option described below. Valid values for <type> include "X", "udp", "tcp", or the numeric value of the IP protocol type of interest. The "X" value "wildcards" the <type> so that <i>trpr</i> will automatically create a plot on the graph for any type of IP protocol which meets the given <source,destination> criteria. The source and destination addresses (srcAddr & dstAddr) must be given in dotted decimal notation or may also be wildcarded with an "X" character. The <source,destination> portion may also be omitted and then will be automatically wildcarded. The optional "id" portion of the flow description corresponds to any "flow id" which may apply to the data analyzed. This currently only applies to <u>drec</u> log files when the user wishes to additionally differentiate <u>drec</u> flows by their "flow id". (See the <i>mgen</i> user's guide for more information). As an example, " auto udp" will cause <i>trpr</i> to enumerate individual plots for each unique UDP protocol flow detected regardless of source or destination. The source and destination port numbers can be explicitly specified or wildcarded with an "X"

or implicitly through omission. Note that flows which match those given with the flow option (see below) will not be tested against the auto criteria. The auto option may be used multiple times on the *trpr* command line to establish multiple sets of automatic flow matching criteria (e.g. *trpr auto udp auto tcp ...* "). Note that if _no_flow or auto filters are provided, *trpr* runs with a default wildcard enumeration filter of "auto X"

flow

<type,srcAddr:port,dstAddr:port,id>

This command instructs *trpr* to look for and plot specific "flows" which match the given (type,src,dst) criteria. All flows which match the given criteria are accumulated together onto a single plot line. The address and port criteria are given in the same way as for the auto command and may be wildcarded in the same way. For example, the option "flow udp" will cause *trpr* to accumulate all detected UDP traffic (regardless of source and destination since they are implicitly wildcarded here) into a single plot. Thus the command "*trpr flow udp flow tcp ...*" will produce a graph with two lines, one plotting cumulative UDP traffic and the other plotting cumulative TCP traffic detected by *tcpdump*. As with the auto option, the flow option may be used multiple times on the command line and may be used in conjunction with the auto option. Flows of network traffic matching the criteria specified with the flow option will be accumulated into a matching flow plot and are also tested against the sets of auto option criteria so redundant plot lines may result depending on the criteria used.

exclude

<type,srcAddr:port,dstAddr:port,id>

This command instructs *trpr* to ignore specific "flows" which match the given (type,src,dst) criteria. The address and port criteria are given in the same way as for the auto command and may be wildcarded in the same way. For example, the option "flow udp" will cause *trpr* to ignore all detected UDP traffic (regardless of source and destination since they are implicitly wildcarded here). The exclude command filters are evaluated before the auto and flow command filters.

input <inputFile>

This option instructs *trpr* to use the file name given by <inputFile> for input. Otherwise *trpr* looks for input from *stdin*. The expected input format is text output from the *tcpdump* program run with its hexadecimal option (-x) given and properly filtered so that only IP protocol data is captured. Non-IP data from *tcpdump* will result in errors in *trpr*'s output.

output <outputFile>

This option instructs *trpr* to save cumulative data into the file name given by <outputFile> for later (non-real-time) plotting. The plot data stored here contains data from the entire *tcpdump* run (as opposed to the *trpr* real-time mode's limited history of data). By default (i.e. unless the raw option is given), the output file contains text header information at its beginning so that gnuplot can be used to create a nicely-labelled graph.

link <src>[,<dst>]

This causes *trpr* to process only packets associated with the identified "link" or "node". For *ns* trace files, the <src> and <dst> values correspond to simulation node identifiers. For *tcpdump* operation, the MAC address is used. Note that <src> and/or <dst> values can be wildcarded by omission or by designating 'X' as the value. For *ns* simulations using the wireless/mobility extensions, the <dst> value may be "AGT"

	<p>or "RTR" corresponding to the wireless transmission type (By default, both "AGT" and "RTR" are counted by <i>trpr</i>) since the notion of "links" is not used in the trace files.</p> <p>Wildcarding the <src> or <dst> values allows the user to analyze all traffic arriving to and/or leaving from a specific simulation node or MAC address. The <u>send</u> and <u>recv</u> commands may be optionally used in combination with the <u>link</u> command to specify whether only arriving packets (<u>recv</u>) or departing packets (<u>send</u>) are processed. By default, both arriving and departing packets are processed.</p>
send	<p>Specifies that only "sent" packets are to be processed. In <i>ns</i>, this corresponds to 's' events for traced links or nodes. In <i>tcpdump</i>, this corresponds to packets whose source MAC address correspond to the <src> value given with the <u>link</u> command. By default, both "sent" and "received" packets are counted by <i>trpr</i>. The <u>send</u> and <u>recv</u> commands are generally useful only for <i>ns</i> simulations but may be applicable to <i>tcpdump</i> trace file analysis in some situations.</p>
recv	<p>Specifies that only "received" packets are to be processed. In <i>ns</i>, this corresponds to 'r' events for traced links or nodes. In <i>tcpdump</i>, this corresponds to packets whose destination MAC address corresponds to the <dst> value given with the <u>link</u> command. By default, both "sent" and "received" packets are counted by <i>trpr</i>. The <u>send</u> and <u>recv</u> commands are generally useful only for <i>ns</i> simulations but may be applicable to <i>tcpdump</i> trace file analysis in some situations.</p>
range <startSec>[:stopSec]	<p>Causes <i>trpr</i> to skip ahead to the "start time" (in seconds) from the first packet event in the trace file and end processing at the optional "stop time" (in seconds). Setting the "stop time" to -1 causes <i>trpr</i> to process until the end of the trace input. Note the <u>range</u> command may be used in combination with the <u>offset</u> and/or <u>absolute</u> commands to perform analysis for a specific time period in the trace file.</p>
offset <hh:mm:ss>	<p>This allows the user to specify an absolute analysis start time using a time-of-day reference. The time given is in 24-hour clock time format and must be within 12 hours of the time of the first packet event in the trace file.</p>
absolute	<p>Causes <i>trpr</i> to use the absolute time given in the trace file in its output instead of "normalizing" the time values (generally the plots' x-axis) to zero time for the first packet event or optional <u>offset</u> time.</p>
summary	<p>This causes <i>trpr</i> to output summary statistics of results to <i>stdout</i> at the end of analysis. These summary results are available with or without the production of data intended for plotting. This options is useful for commanding or scripting <i>trpr</i> to collect statistics in addition to or instead of plots.</p>
histogram	<p>This causes <i>trpr</i> to output a histogram of the values of analyses intervals (intervals determined by the window command) for each flow to <i>stdout</i>. Some percentile information of the histogram content is also provided in the output. The histograms are comma-delimited tables of values. The <i>hcat</i> program provided in the TRPR distribution can be used to query and manipulate these histogram files or they can also be plotted with a graphing tool (e.g. <i>gnuplot</i>). The <i>hcat</i> program also allows multiple histogram files from multiple <i>trpr</i> analysis runs to be combined together for cumulative statistics collection. Currently the quantization size and curve</p>

of the histogram is fixed and adapts in range with data. The histogram output may be useful for packet latency analyses or other kinds of statistics compilations.

This limits *trpr*'s rate of real-time *gnuplot* graph generation to a <factor> of real time when parsing a pre-existing trace file. When the replay command is given, *trpr* generates the same *gnuplot* output as for the real command. The <factor> parameter scales the playback rate with respect to real time. For example, <factor> = 1 is actual real time, while <factor> = 2 is double speed playback. Note that real time update occurs once per window time.

replay <factor>

***tcpdump* Hints:**

- 1) Make sure you are using *tcpdump* filters such that only IP packets are captured (*trpr* currently doesn't like non-IP packets in *tcpdump* 's output).
- 2) Always use the "-x" option when using *tcpdump* with *trpr*. (*trpr* looks for and parses the hexadecimal output)
- 3) Use *tcpdump*'s "-n" option to skip DNS lookups and speed up *tcpdump*'s performance (*trpr* only uses dotted decimal numeric IP addresses).
- 4) Use *tcpdump*'s line buffering option ("-1") to get output with minimal delay for real time plotting.
- 5) Read and learn *tcpdump*'s man page for the extensive set of filtering options *tcpdump* provides. Uses these filter options in conjunction with *trpr*'s own filters to get the graphical results you want.
- 6) Leverage *tcpdump*'s ability to store captured data in a binary file (use *tcpdump*'s "-w" option) and then post-process it with *tcpdump* 's filter's (using *tcpdump* to process the stored binary file with its "-r" option and redirecting its output to *trpr*).

***gnuplot* Hints:**

- 1) Use *gnuplot*'s "-noraise" option when using with *trpr* in "real-time" mode if you don't want the updated plots to continually pop to your display's top level.
- 2) Use *gnuplot*'s "-persist" option if you wish the last plot to remain displayed after exiting.
- 3) *trpr*'s output files for *gnuplot* are in text format and easily edited to customize output. *Gnuplot* is a very flexible program with lots of options to get the graphs into almost any format you would like. It is also lightning fast.

Examples of Use:

(TBD)

Credits:

trpr was written by:

Brian Adamson
adamson@itd.nrl.navy.mil

Your questions and comments are welcome and appreciated. (16 April 2004)